Multi-objective evolutionary–fuzzy augmented flight control for an F16 aircraft

P Stewart[1*], **D Gladwin**[2], **M Parr**[2], and **J Stewart**[1]
[1]School of Engineering, University of Lincoln, UK
[2]Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield, UK

The manuscript was received on 30 April 2009 and was accepted after revision for publication 16 September 2009.

DOI: 10.1243/09544100JAERO610

Abstract: In this article, the multi-objective design of a fuzzy logic augmented flight controller for a high performance fighter jet (the Lockheed-Martin F16) is described. A fuzzy logic controller is designed and its membership functions tuned by genetic algorithms in order to design a roll, pitch, and yaw flight controller with enhanced manoeuverability which still retains safety critical operation when combined with a standard inner-loop stabilizing controller. The controller is assessed in terms of pilot effort and thus reduction of pilot fatigue. The controller is incorporated into a six degree of freedom motion base real-time flight simulator, and flight tested by a qualified pilot instructor.

Keywords: flight simulation, multi-objective design, fuzzy control optimization genetic algorithms

1 INTRODUCTION

This article addresses the central premise of designing a stable yet aggressive flight controller for a high performance manned fighter aircraft, viz., *The Lockheed-Martin F16 Fighting Falcon* jet fighter. This aircraft was one of the first widespread implementations of fly-by wire and represented at the time of its introduction a step change in terms of its inherently unstable airframe and computer flight controller. Of particular interest in terms of pilot efficiency is the amount of pilot effort in terms of corrections which have to be input to the flight controls to achieve trajectory tracking.

The F16 flight dynamics are non-linear, time varying, and traditionally flight controllers have been designed by utilizing linearized aircraft models operating at a variety of set-point operating conditions [**1**]. These linearized, time-invariant models yield gain-scheduled controllers, which do not exhibit globally acceptable performance [**2**]. In order to address these problems, non-linear techniques such as feedback linearization have become a de-facto standard as an alternative to gain scheduling [**3**]. In this article, an existing feedback linearization scheme is augmented by a fuzzy logic controller that has been designed and optimized by a multi-objective evolutionary algorithm (MOEA) in order to take advantage of the inherently stable aspects of the feedback linearized controller, merged with the aggressive flight control aspects of the fuzzy controller.

Good performance has been reported using a variety of techniques, such as robust dynamic inversion [**4**, **5**], neural networks [**6**, **7**], Lyapunov functions [**8**], receding horizon control [**9**], and fuzzy logic [**8**, **10**, **11**]. Although these approaches give good results, some either utilize simplifications or linearizations related to the aircraft model, not encompassing the non-linear six-degrees-of-freedom (6DOF) aircraft dynamics [**7**]. Recently, non-linear techniques have been extended some way to 6DOF dynamics [**12**], complex multiple structure design techniques [**13**] (which only deals with one longitudinal channel), non-linear neural network-based control [**14**], which addresses stability in multiple axes and uncertain aircraft dynamics.

Finally, it has been recognized [**15**] that a holistic approach to design incorporating artificial intelligence (AI) methodologies such as fuzzy logic, combined with techniques capable of incorporating multiple and often conflicting objectives and constraints

*Corresponding author: *School of Engineering, University of Lincoln, Brayford Pool, Lincoln LN6 7TS, UK.*
email: pstewart@lincoln.ac.uk

has the potential to deliver high performance control not only in multiple physical axes but also multi-dimensional objective solution spaces.

In this article, a comprehensive design approach is proposed, based around a non-linear 6DOF F16 model, which is controlled by an existing feedback linearized 6DOF flight controller. The objective is to design an augmented controller that delivers aggressive aerobatics when compared with the linearized controller, while still retaining its associated stable operation. Although flight control and fighter manoeuvre have previously been considered using radial basis function networks and Lyapunov functions [**8**, **16**], a degree of training and learning is required to develop both model and controller. In order to develop a more generic approach and avoid the need for training, in this article, a fuzzy logic controller is developed on a real-time authentic aircraft model, and is tested by a qualified pilot instructor against a realistic flight plan.

The augmented controller was developed offline in *Simulink*, then integrated into the real-time operating system of a 6DOF flight simulator in the Faculty of Engineering at the University of Sheffield (Fig. 1).

The Simulator is an *Explorer RD* manufactured by CueSim Ltd. (www.cuesim.co.uk). The aerodynamic data used in the simulation was derived from NASA Langley wind-tunnel tests conducted on a scale model of an F-16 fighter [**17**], and model by Stevens and Lewis [**18**].

2 AIRCRAFT 6DOF EQUATIONS AND MODEL

2.1 Aircraft equations of motion

The 6DOF equations of motion for an aircraft (assuming a flat non-rotating earth) are given by [**16**]

$$\dot{u} = rv - qw + \frac{\bar{q}S}{m}C_x - g\sin\theta + \frac{T}{m} \tag{1}$$

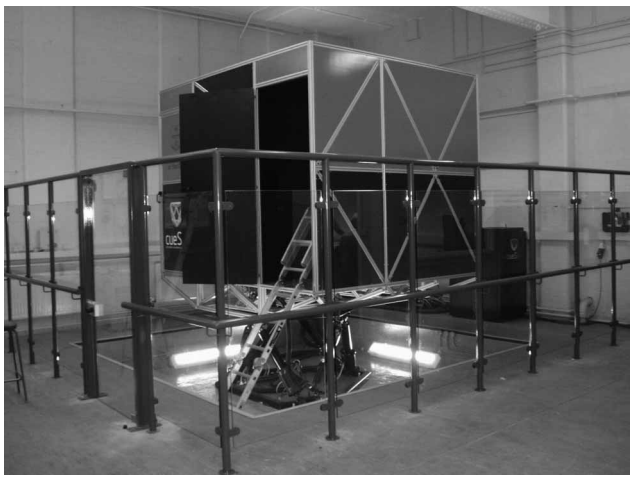

Fig. 1 University of Sheffield Faculty of Engineering Flight Simulator

$$\dot{v} = pw - ru + \frac{\bar{q}S}{m}C_y + g\sin\phi\cos\theta \tag{2}$$

$$\dot{w} = qu - pv + \frac{\bar{q}S}{m}C_z + g\cos\phi\cos\theta \tag{3}$$

$$\dot{p} = (J_1 r + J_2 p)q + J_3 L + J_4 N \tag{4}$$

$$\dot{q} = J_5 pr - J_6(p^2 - r^2)q + J_2 M \tag{5}$$

$$\dot{r} = (J_8 p - J_2 r)q + J_4 L + J_9 N \tag{6}$$

$$\dot{\phi} = q\tan\theta\sin\phi + r\tan\theta\cos\phi \tag{7}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{8}$$

$$\dot{\psi} = r\cos\phi\sec\theta + q\sin\phi\sec\theta \tag{9}$$

$$\begin{aligned}\dot{P}_{\mathrm{N}} = {} & u\cos\theta\cos\psi + v(-\cos\phi\sin\psi \\ & + \sin\phi\sin\theta\cos\psi) + w(\sin\phi\sin\psi \\ & + \cos\phi\sin\theta\cos\psi)\end{aligned} \tag{10}$$

$$\begin{aligned}\dot{p}_{\mathrm{E}} = {} & u\cos\theta\sin\psi + v(\cos\phi\cos\psi \\ & + \sin\phi\sin\theta\sin\psi) + w(-\sin\phi\cos\psi \\ & + \cos\phi\sin\theta\sin\psi)\end{aligned} \tag{11}$$

$$\dot{h} = u\sin\theta - v\sin\phi\cos\theta - w\cos\phi\sin\theta \tag{12}$$

where (u, v, w) are the velocities in the $(X_{\mathrm{B}}, Y_{\mathrm{B}}, Z_{\mathrm{B}})$ body axes, ϕ is the roll angle, θ is the pitch angle, ψ is the yaw angle, p is the roll rate, q is the pitch rate, r is the yaw rate, p_{N} is the north position, p_{E} is the east position, and h is the altitude. The moments of inertia are

$$\Gamma = I_X I_Z - I_{XZ}^2 \tag{13}$$

$$J_1 = (I_Y I_Z - I_Z^2 - I_{XZ}^2)/\Gamma \tag{14}$$

$$J_2 = (I_X - I_Y + I_Z)I_{XZ}/\Gamma \tag{15}$$

$$J_3 = I_Z/\Gamma \tag{16}$$

$$J_4 = I_{XZ}/\Gamma \tag{17}$$

$$J_5 = (I_Z - I_X)/I_Y \tag{18}$$

$$J_6 = I_{XZ}/I_Y \tag{19}$$

$$J_7 = 1/I_Y \tag{20}$$

$$J_8 = \left[I_X(I_X - I_Y) + I_{XZ}^2\right]/\Gamma \tag{21}$$

$$J_9 = I_X/\Gamma \tag{22}$$

where (I_X, I_Y, I_Z) are the moments of inertia of the body axis system and I_{XZ} is the $(X_{\mathrm{B}} - Z_{\mathrm{B}})$ body axis product of inertia.

The aerodynamic forces $(X_{\mathrm{A}}, Y_{\mathrm{A}}, Z_{\mathrm{A}})$ and moments $(L_{\mathrm{A}}, M_{\mathrm{A}}, N_{\mathrm{A}})$ act on the aircraft via dimensionless aerodynamic force and moment coefficients

$$X_{\mathrm{A}} = \bar{q}SC_x$$

$$Y_{\mathrm{A}} = \bar{q}SC_y$$

$$
\begin{aligned}
Z_{\mathrm{A}} &= \bar{q}SC_z \\
L_{\mathrm{A}} &= \bar{q}SbC_{\mathrm{l}} \\
M_{\mathrm{A}} &= \bar{q}S\bar{c}C_m \\
N_{\mathrm{A}} &= \bar{q}SbC_n
\end{aligned}
\tag{23}
$$

where $\bar{q}$ is the dynamic pressure, S is the reference area, $\bar{c}$ is the average geometric chord, b is the reference span, C_x is the x body-axis aerodynamic force coefficient, C_y is the aerodynamic side-force coefficient, C_z is the z body-axis aerodynamic force coefficient, and (C_l, C_m, C_n) are the moments for aerodynamic roll, pitch, and yaw, respectively.

The total aircraft velocity (V), angle of attack (α), and side-slip angle (β) are derived as

$$\dot{V}_{\mathrm{T}} = (u\dot{u} + v\dot{v} + w\dot{w})/V_{\mathrm{T}} \tag{24}$$

$$\dot{\beta} = (\dot{u}V_{\mathrm{T}} - v\dot{V}_{\mathrm{T}})/(V_{\mathrm{T}}^2 \cos\beta) \tag{25}$$

$$\dot{\alpha} = (u\dot{w} - w\dot{u})/(u^2 + w^2) \tag{26}$$

$$u = V_{\mathrm{T}} \cos\alpha \cos\beta \tag{27}$$

$$v = V_{\mathrm{T}} \sin\beta \tag{28}$$

$$w = V_{\mathrm{T}} \sin\alpha \cos\beta \tag{29}$$

2.2 Aircraft model

The motion of the aircraft is represented by the vector differential equation [**18**]

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(C, \boldsymbol{x}, \boldsymbol{u}) \tag{30}$$

where $\boldsymbol{f}$ is the vector function and ($\boldsymbol{x}$, $\boldsymbol{u}$, C) represent the state vector, input vector, and dynamic coefficients. The states describing the rigid body motion of an aircraft with respect to a flat earth are

$$X^{\mathrm{T}} = [V, \alpha, \beta, \phi, \theta, \psi, p, q, r, p_{\mathrm{N}}, p_{\mathrm{E}}, h] \tag{31}$$

where V is the total aircraft velocity, α is the angle of attack, β is the sideslip angle, ϕ is the pitch angle, ψ is the yaw angle, p is the roll rate, q is the pitch rate, r is the yaw rate, p_{N} is the north position, p_{E} is the east position, and h is the altitude. The input vector of the model is

$$\boldsymbol{U}^{\mathrm{T}} = [\delta_{\mathrm{e}}, \delta_{\mathrm{a}}, \delta_{\mathrm{r}}, \delta_{\mathrm{t}}] \tag{32}$$

where δ_{e} is the elevator deflection, δ_{a} is the aileron deflection, δ_{r} is the rudder deflection, and δ_{t} is the throttle setting.

The F-16 engine power response is modelled as a first-order lag. The engine module contains thrust information as a function of throttle setting, Mach number and altitude. The effect of fuel burn is modelled to allow a more realistic mission simulation.

The aerodynamic data are derived from F-16 scale model wind-tunnel tests [**17**]. The body-axis aerodynamic coefficients are held in multi-dimensional look-up tables with associated linear interpolation algorithms. The data covers an angle of attack range from $-20°$ to $90°$, and a range of sideslip angles from $-30°$ to $30°$. The deflection limit of the elevator is $\pm 25°$, the ailerons is $\pm 21.5°$, and the rudder is $\pm 30°$.

2.3 Inner control loop

A standard inner-loop stabilizing controller had previously been designed and integrated into the F-16 simulator [**4**], which modifies the plant dynamics under varying flight conditions using the dynamic inversion method.

The controlled variables for pitch, roll, and yaw are denoted as M_{cv}, L_{cv}, and N_{cv} respectively. The controlled variable for the longitudinal dynamics is a combination of pitch rate and angle of attack

$$M_{\mathrm{cv}} = q + 0.1\alpha \tag{33}$$

and the variables for the roll and yaw moments L and N are

$$
\begin{aligned}
L_{\mathrm{cv}} &= p \\
N_{\mathrm{cv}} &= r
\end{aligned}
\tag{34}
$$

The generalized controls are thus

$$
\begin{aligned}
\dot{M}_{\mathrm{cv(cmd)}} &= \dot{q}_{\mathrm{cmd}} + 0.1\dot{\alpha}_{\mathrm{cmd}} \\
\dot{L}_{\mathrm{cv(cmd)}} &= \dot{p}_{\mathrm{cmd}} \\
\dot{N}_{\mathrm{cv(cmd)}} &= \dot{r}_{\mathrm{cmd}}
\end{aligned}
\tag{35}
$$

The open loop plant to be inverted is

$$\dot{M}_{\mathrm{cv}} = [M_\alpha + 0.1Z_\alpha \quad M_q + 0.1Z_q][\alpha \quad q]^{\mathrm{T}} + \dot{M}_{\mathrm{cv(cmd)}} \tag{36}$$

$$\begin{bmatrix} \dot{L}_{\mathrm{cv}} \\ \dot{N}_{\mathrm{cv}} \end{bmatrix} = \begin{bmatrix} L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{L}_{\mathrm{cv(cmd)}} \\ \dot{N}_{\mathrm{cv(cmd)}} \end{bmatrix} \tag{37}$$

For dynamic inversion, the state equation is manipulated to solve for the control thus

$$
\begin{aligned}
\dot{y}_{\mathrm{d}} &= h(x) + g(x)u \\
g(x)u &= \dot{y}_{\mathrm{d}} - h(x) \\
u &= g(x)^{-1}(\dot{y}_{\mathrm{d}} - h(x))
\end{aligned}
\tag{38}
$$

where y_d is the desired output dynamics, $h(x)$ is the plant dynamics, and $g(x)$ is the controller effectiveness. If the desired dynamics are written as a linear function of the states, then since $g(x)$ is an identity, u becomes

$$u = (A_d - A)x \tag{39}$$

The closed loop longitudinal dynamics are chosen such that the aircraft will maintain an equilibrium position. The choice of dynamics is

$$\dot{M}_{cv_{desired}} = [-0.1 \quad -3] \begin{bmatrix} \alpha \\ q \end{bmatrix} \tag{40}$$

The lateral dynamics are designed by applying eigenstructure theory to the three-state model [**19**]. The central operational design point is located at an altitude of 2000 ft, at Mach 0.5 and has desired poles of -2.5 ± 2.0i and -3.0. The second and third rows of the closed loop **A** matrix, which are used as $\mathbf{A}_d$ are

$$\mathbf{A}_d = \begin{bmatrix} -0.083 & -3.006 & 0.049 \\ 12.029 & 0.429 & -4.726 \end{bmatrix} \tag{41}$$

Therefore, the inverse dynamics can be solved for all three axes

$$\dot{M}_{cv} = \begin{bmatrix} -0.1 - (M_\alpha + 0.1Z_\alpha) & -3.0 - (M_q + 0.1Z_q) \end{bmatrix} \times \begin{bmatrix} \alpha \\ q \end{bmatrix} \tag{42}$$

$$\begin{bmatrix} \dot{L}_{cv(cmd)} \\ \dot{N}_{cv(cmd)} \end{bmatrix} = \begin{bmatrix} -0.083 - L_\beta & -3.006 - L_p & 0.048 - L_r \\ 12.029 - N_\beta & 0.429 - N_p & -4.726 - N_r \end{bmatrix} \times \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} \tag{43}$$

The (M, L, N) derivatives are stored as lookup table, which is retrieved to calculate the inner loop commands.

3 FUZZY LOGIC CONTROLLER

Upon consideration of the number of flight performance variables related to the calculation of the final flight surface position for any pilot input demands, a fuzzy logic flight management controller was developed. This management controller would assess the pilot's demands and apply rules to create an output that utilizes the full potential of the F-16's '*relaxed static stability*' [**20**]. The final output would be '*defuzzified*' by a real-time updated output membership set calculated from the flight performance variables to give optimum flight surface deployment values for that instance. The fuzzy logic controller was designed through a process of five steps:

(a) input and control variables definition;
(b) condition interface definition;
(c) rule-base design;
(d) computational unit design;
(e) de-fuzzification design.

3.1 Input, control variables and condition interface

The input variables were selected as the pilot pitch, roll, and yaw rate commands as well as the corresponding demand rate-of change. The control variables were the aileron, elevator, and rudder output demands to the physical control section of the F-16 model. The condition interface is the method in which observations of the processes are expressed as members of the fuzzy linguistic sets.

The pitch, roll, and yaw membership sets were constructed in an identical manner, although the range the membership sets covered was altered according to the relative range of the input signals. Representative function plots for variable 'roll' are shown in Figs 2 to 4. Gaussian type membership function shapes have been chosen due to their smooth, nonlinear differentiable functions, which are thought to

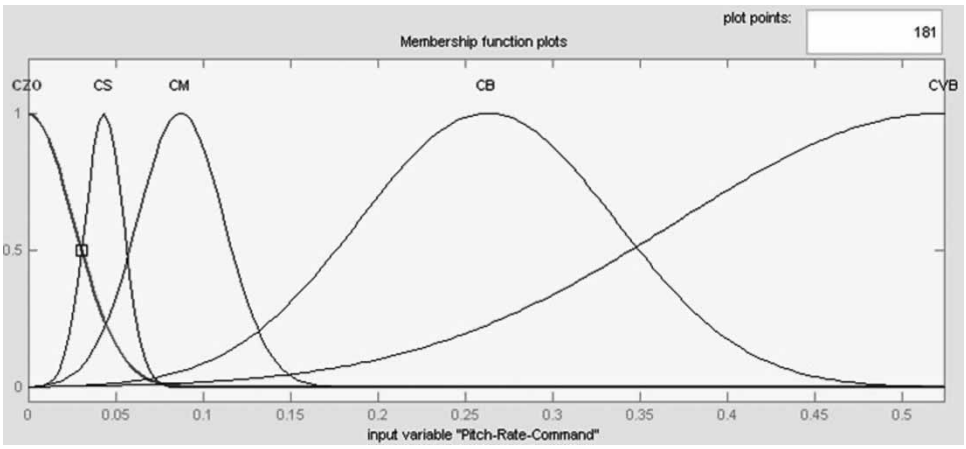

Fig. 2 Roll rate command function

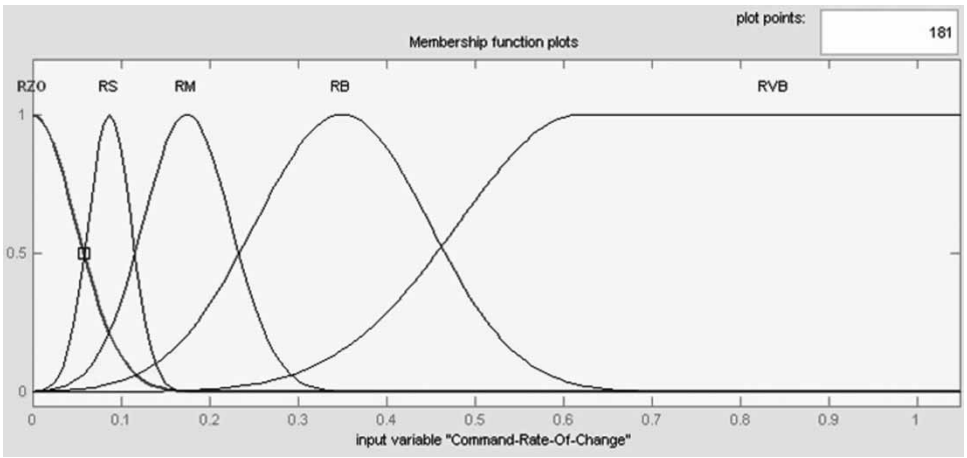

Fig. 3 Rate of change of roll rate function

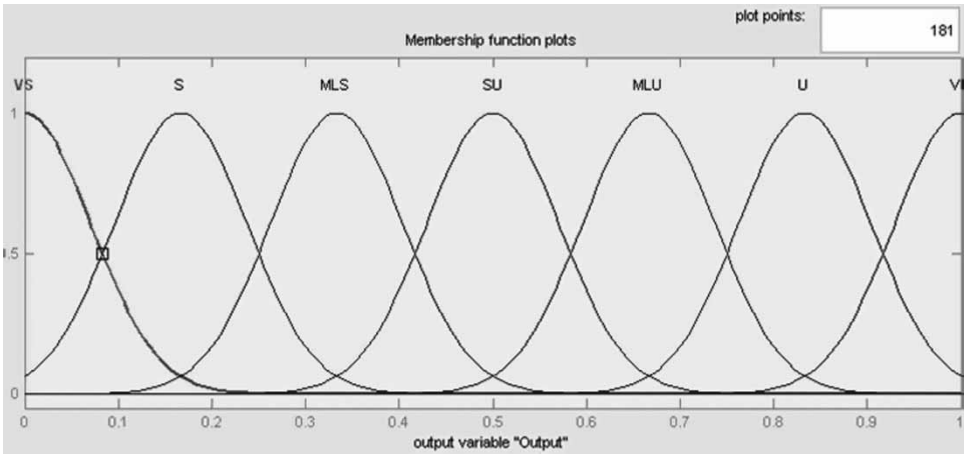

Fig. 4 Roll output function

represent human experience more accurately than triangular/trapezoidal function, which are often chosen for computational efficiency rather than function. Several studies have been conducted to compare the highly subjective performance of membership function shapes, which is beyond the scope of this article [**21**, **22**].

3.2 Rule base

There were a number of specifications for any designed control system. In particular, the objectives for the new flight controller were defined as follows.

1. The aircraft should have acceptable flying qualities as defined using the Cooper–Harper pilot rating scale [**23**].
2. The quality of the controller must be an improvement on the original controller.
3. The aircraft should respond precisely to small absolute stick movements.
4. The aircraft should respond rapidly to large rate of change stick demands.
5. Low phase lag between cockpit controller and flight control surface.
6. The controller must not allow the aircraft to become unstable.
7. The controller must not allow the aircraft to exceed its safe characteristics (i.e. Max G).

A number of these objectives can be considered to be subjective and therefore a trained pilot was used in consultation to help decide, in the final assessment of the controller, whether they had been met. The rule base uses the linguistic sets defined by the inputs to determine the optimum output. The design of the rule base consists of determining which rule should be applied under which circumstance. To help decide in this matter a controller specification was developed. The rule base was designed to operate to produce output results in the manner shown in Fig. 5.

The relational surface for the motion controllers is shown in Fig. 6.

Output Flight Surface Response		Input Rate Demand				
		Zero Order	Small	Medium	Large	Very Large
Input Acceleration Rate	Zero Order	VS	VS	S	SU	SU
	Small	VS	S	S	MLS	MLU
	Medium	S	S	MLS	MLU	U
	Large	SU	MLS	MLU	U	U
	Very Large	SU	MLU	U	U	VU

Designed rule base

where VS – Very Stable
S – Stable
MLS – More or Less Stable
SU – Stable-Unstable
VU – Very Unstable
U - Unstable
MLU – More or Less Unstable

Fig. 5 Rule base

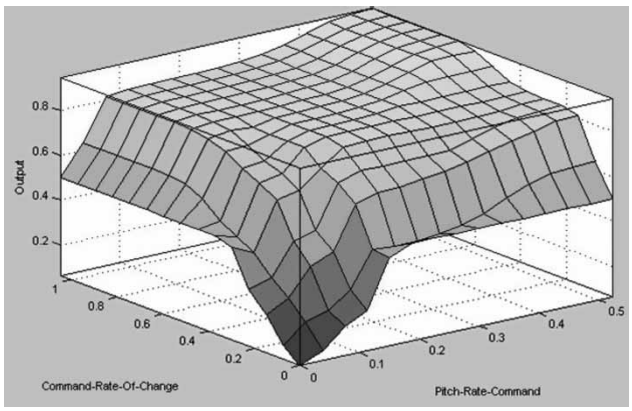

Fig. 6 Relational surface

3.3 De-fuzzification

The computational unit utilized Mamdani-type fuzzy processing [**5**]. The computational unit uses the centre of area (COG) method to '*de-fuzzify*' the results and provide a scale factor applied to the raw pilot input demands and the original controller action to give the desired action. The COG method has both continuity and un-ambiguity, which were considered more desirable than the simpler computational complexity of the mean of maxima method. The scale factor produced by the fuzzy logic flight management controller ranged between one and zero. An output of zero was considered very stable and equivalent to the original controller. An output of one was considered very unstable and equivalent to there being no flight controller present at all; this was simulated by passing the pilot inputs directly through as surface demands to the physical model.

The testing on the explorer RD demonstrated the possibility of G-forces being placed on the aircraft which exceeded the maximum the plane structure could accommodate. The specification specifically did not allow this and for this reason management of the G-forces experienced was included in the new flight controller. If the aircraft started to approach the maximum or minimum G-forces allowed, the flight controller would demand the aircraft to respond in a completely stable manner to prevent any structural damage to the aircraft. The controller was tested by inputting small (10 per cent of full demand) and large (95 per cent of full demand) disturbances at low (Mach 0.25) and high (Mach 1.75) speeds to the system with and without the new flight management controller and analysing the results.

The results demonstrated that while the aircraft manoeuvrability may have been increased, its performance in tracking pilot demands was poor and in most cases poorer than the original F-16 flight controller; showing evidence of high overshoot and large settling times.

In response to the rather poor performance of the controller, there are a number of options to be considered, a notable example of which is the Takagi–Sugeno modelling approach, which has been demonstrated with excellent results in the control of non-linear systems [**24**–**26**]. However, to improve the performance of the controller it was decided to use genetic algorithms (GAs) to tune the fuzzy controller [**27**], which has a number of advantages. GAs are ideally suited to tune controllers for multi-input multi-output (MIMO) systems, and in particular, the multi-objective genetic algorithm (MOGA) has been shown to deal effectively with multi-variable, competitive and complex objective functions that have robust characteristics [**28**]. As the problem under consideration here is MIMO, has a competitive, multiple objectives, and will ideally have some aspects of robustness in performance, the choice of MOGA as a tuning tool for the fuzzy logic controller was confirmed.

GAs are a stochastic global search and optimization method that mimic natural biological evolution through the principles of natural selection, genetic modification and selective breeding. GAs operate on a population of individuals with each individual representing a possible solution to the optimization problem.

GAs begin with the random initiation of a population. With each generation of a GA, a new set of approximations is created by a process of selection, crossover and mutation. The selection process determines the fittest individuals to go on to the next population. Crossover exchanges the genetic material of two of individuals, creating two new individuals. Mutation changes, at random, the genetic material of an individual. This process leads to the evolution of a population that is better suited to its environment than the individuals from which it was created, just as with natural adaptation.

The solution is usually achieved when a certain number of generations have been reached. GAs are applicable to non-linear optimization problems that make them ideal for optimally tuning the fuzzy logic flight management controller.

The GA was written in Matlab incorporating the MOGA toolbox [**29**, **30**] and integrating the full F-16 system model including the new fuzzy logic flight management controller [**28**]. The fuzzy logic controllers for pitch, roll, and yaw were each optimized individually using MOGA techniques. It was decided to optimize the membership sets of each controller to attempt to improve the overall system performance. The initial population of individuals was selected at random but within boundaries stipulated by the pilot demands. The population was limited to a hundred individuals to allow for a large spread of values but not increase the processing time significantly. The variables were defined as the parameters of each class in the membership sets. Each class required two parameters to define its Gaussian shape; a location point for the peak value and a width. With 17 classes contained in the three membership sets (*the G-Force membership set was not optimized as this was considered safety critical*) thus the MOGA optimized 34 variables. This phase of the algorithm applies the individuals to the objective functions and measures how well they complete the task. The level to which the individual successfully completed the task is known as its fitness.

3.4 MOGA overview

Formally, and without loss of generality [**31**], multi-objective optimization can be expressed as

$$\text{Minimize } \boldsymbol{f}(x)$$

where $\boldsymbol{f}(x) = (f_1(x), \ldots, f_n(x))$ is a vector of objective functions, n is the number of objectives or criteria to be considered, $\boldsymbol{x} = (x_1, \ldots, x_p)$ is a vector of decision variables, and p is the number of decision variables that comprise the complete solution. In the absence of preference information, solutions to multiobjective problems are compared using the notion of *Pareto dominance*. A particular solution x, with associated performance vector $\boldsymbol{u}$, is said to dominate, another solution y with performance vector $\boldsymbol{v}(x, p, y)$ if the former performs at least as well as the latter across all objectives, and exhibits superior performance in at least one objective. A solution is said to be *Pareto optimal* if it is not dominated by any other possible solution. The Pareto-front is the set of points in criterion-space that correspond to the Pareto-optimal solutions. Without *a priori* or progressive preference articulation, a multi-objective search engine will generally aim to discover a family of solutions that provide a good representation of the Pareto front (Figs 7 and 8).

The first Pareto-based MOEA to be published was the MOGA [**29**]. GAs are suitable search engines for multi-objective problems primarily because of their population-based approach. An MOEA is capable of supporting diverse, simultaneous, solutions in the search environment. A carefully designed GA is robust in the face of ill-behaved cost landscapes featuring attributes such as multi-modality and discontinuity. Furthermore, the GA methodology offers a flexible choice of decision variables and objective

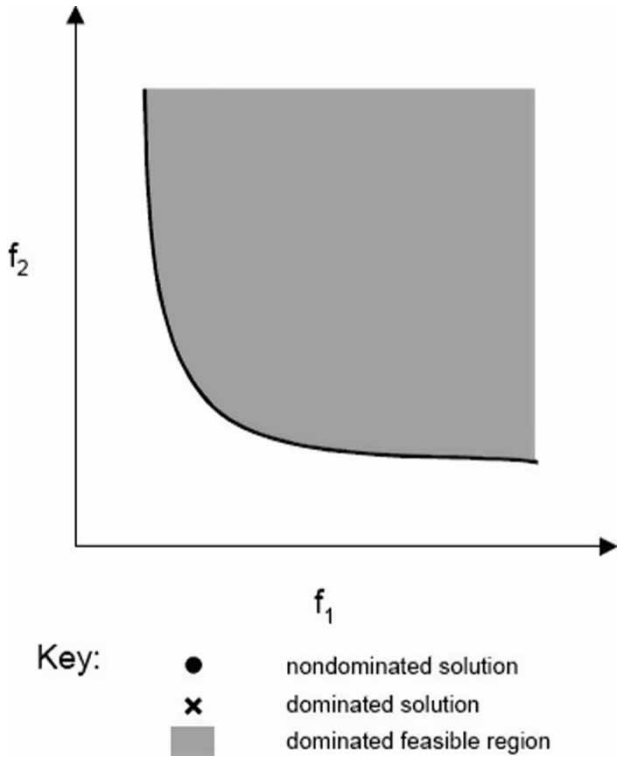

Fig. 7 Non-dominated solution

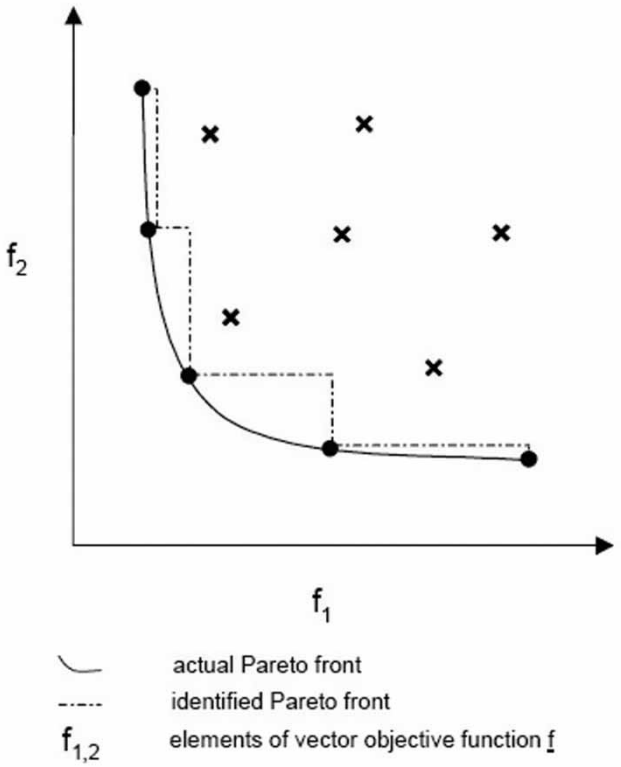

Fig. 8 Pareto optimality

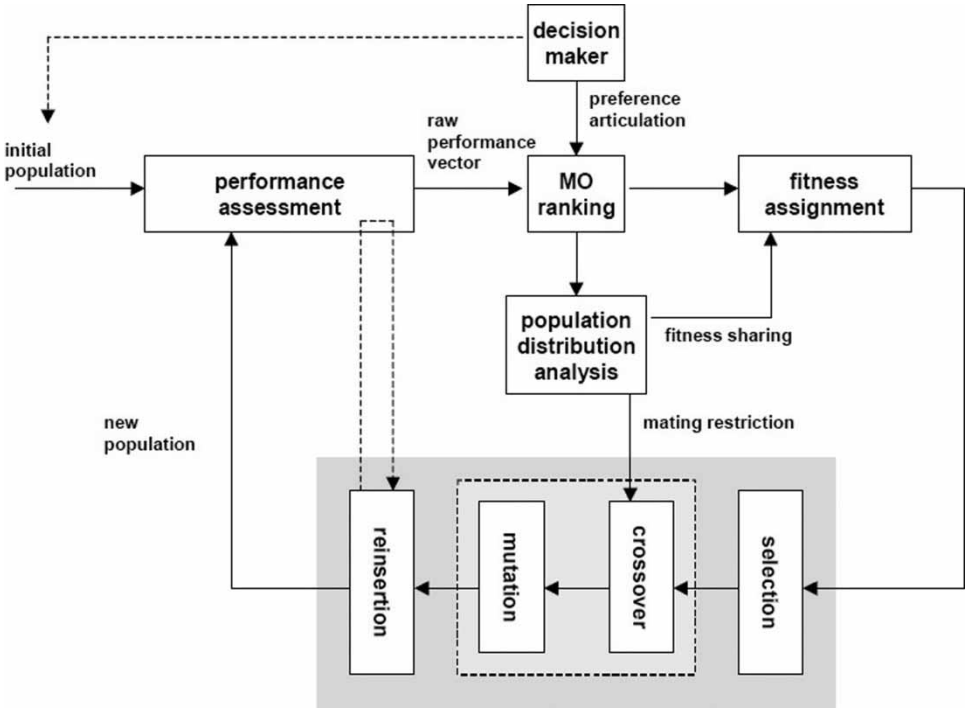

Fig. 9 MOGA schematic

specifications. A general schematic of the MOGA is shown in Fig. 9. The MOGA framework can be seen to incorporate all the elements of the standard, single objective, GA. A population of potential solutions is instantiated, then assessed and manipulated over a number of iterations in order to obtain a good solution or set of solutions. Performance assessment, selection, genetic operators (such as crossover and mutation), and reinsertion phases are functionally, in a general sense, the same for the MOGA as for the standard GA.

Population distribution analysis, in which a measure of the density of the population is made, has also been applied in the single objective case to cater for multimodal cost landscapes. The results of this analysis are used in niching and mating restriction schemes.

Multi-objective ranking, which impacts primarily on fitness assignment, is the key difference between the MOGA and a standard GA. Interaction with a decision-maker (DM), or group of DMs, is made explicit in Fig. 9. The DM may choose to introduce *a priori* information into the initial population (at the very least, this would include appropriate limits on decision variables), as is sometimes the case in standard GA applications. With the MOGA, the DM can also seek to influence the search while it is in progress by expressing preference for particular solutions or, more generally, the likely attributes of a good solution.

The essential difference between an MOGA and a single objective GA is the method by which fitness is assigned to potential solutions. Each solution will have a vector describing its performance across the set of criteria. This vector must be transformed into a scalar fitness value for the purposes of the GA. This process is achieved by ranking the population of solutions relative to each other, and then assigning fitness based on rank. Individual solutions are compared in terms of Pareto dominance. This notion was introduced into the field of GAs by Goldberg [**32**]. MOGA uses a variation of Goldberg's proposition in order to determine ranks. Each individual is assigned a rank based on the number of individuals by which it is dominated (Fig. 10).

In the absence of preference information, *Pareto dominance* is used to discriminate between two

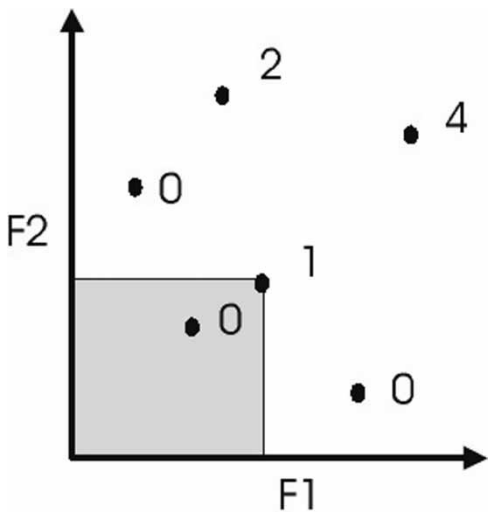

Fig. 10 Multi-objective ranking

competing solutions. However, by involving a DM in the search, other factors can be used to determine superiority. Fonseca and Fleming [**30**] introduced a preferability operator, which discriminates between solutions on the basis of which is preferred by the DM. In Fonseca and Fleming's scheme, the DM can set goal levels and priorities for each of the objectives. These can be refined as the search progresses. This information feeds into the preferability operator, which is used to rank solutions in a similar fashion to the standard Pareto-based approach. Each potential solution is given a rank based on how many other solutions are preferred to it. The preferability operator can be seen as a unification of several popular preference articulation schemes adopted in the wider operational research community. Pareto optimality, the lexicographic method, goal programming, constraint satisfaction, and constrained optimization can all be described by special cases of the preferability operator.

The individuals were assigned a fitness based upon the following four objective functions:

(a) overshoot;
(b) rise-time;
(c) target constraint;
(d) iterative time area error (ITAE).

The target constraint was a check to ensure that the controller constructed using the individual's phenotypes (variables) was possible of a response equal to that of the demand. ITAE [**33**] is a method of analysing the error between the demand signal and the controllers. The integration of time means this objective function especially penalizes large settling times as well as large deviations from the demanded signal. Assigning the fitness required ranking the values based on how well they minimized the objective functions. These functions returned a column vector containing the relevant individual fitness values ranked upon the goals and priorities declared in the MOGA's initialization. The goals and priorities were arranged and ranked as follows:

Priorities

(a) manufacturer's maximum g specification for the airframe (9g for the F16);
(b) feasibility of the solution.

Goals

(a) overshoot;
(b) rise-time;
(c) ITAE;
(d) magnitude of stick movement;
(e) bandwidth of stick movement.

The consequence of assigning priorities is to set hard constraints in the solution space of the search, in order only to return controllers which are physically possible to implement safely on the airframe. The goals were assigned equal significance in the objective function. The nature of Pareto multi-objective optimization is to deliver a 'family' of non-dominated solutions. The assignment of relative merit to the individual components of the objective function can lead to a more homogeneous solution set, but may exclude valid solutions. Hence equal significance was adopted [**30**].

The simulation was run using a modified form of the F-16 Simulink model including the new fuzzy logic flight management controller. The model was modified to record all the required data for objective function analysis and to input step commands of varying magnitude. The pitch, roll, and yaw axis were each tested independently using small, medium, and large pilot rate demands. The initialization files of the model were modified to place the aircraft level at 25,000 ft and airspeed of 500 knots. The F-16 autopilot functions were appropriately used to hold the aircraft in this position except in the axis being tested. The model was run in response to varying step inputs as discussed for a simulation time of 25 s. The individuals in the MOGA were to be used as fuzzy logic controllers in the Simulink model. This portion of the GA chose which values from the population were to be used in the crossover phase. This was achieved based upon the fitness of an individual and the population as a whole. There are a number of different methods to accomplish this; however, stochastic universal sampling was selected for this GA.

Individuals are selected based upon their fitness. The better suited they are to the task, the higher the probability they are to be selected for cross over to produce the next generation.

A generation gap of 80 per cent was used to per cent of the population to continue through to the next generation with no modification. This allows preservation of optimum values from the previous generation to ensure that the new generation's performance does not deteriorate when compared with the previous generation. The MOGA created was used to produce optimum controllers for pitch, roll, and yaw responses to small, medium, and large pilot rate inputs. The results were used to develop optimum pitch, roll, and yaw controllers for all pilot rate demands.

The response surfaces produced by the various fuzzy logic flight management controllers were analysed at their significant areas when considering the pilot demand. This analysis allowed the correct shape of surface required for an optimum controller to be determined and these are shown in Figs 11 to 13.

The analysis of the surfaces produced raised some interesting results. Most notably that the optimization did not recommend a highly unstable response to large pilot demands. The MOGA results did not recommend that the system ever experience instability greater than 60 per cent of the maximum available to prevent large overshoot of the required rate demand. Also the results

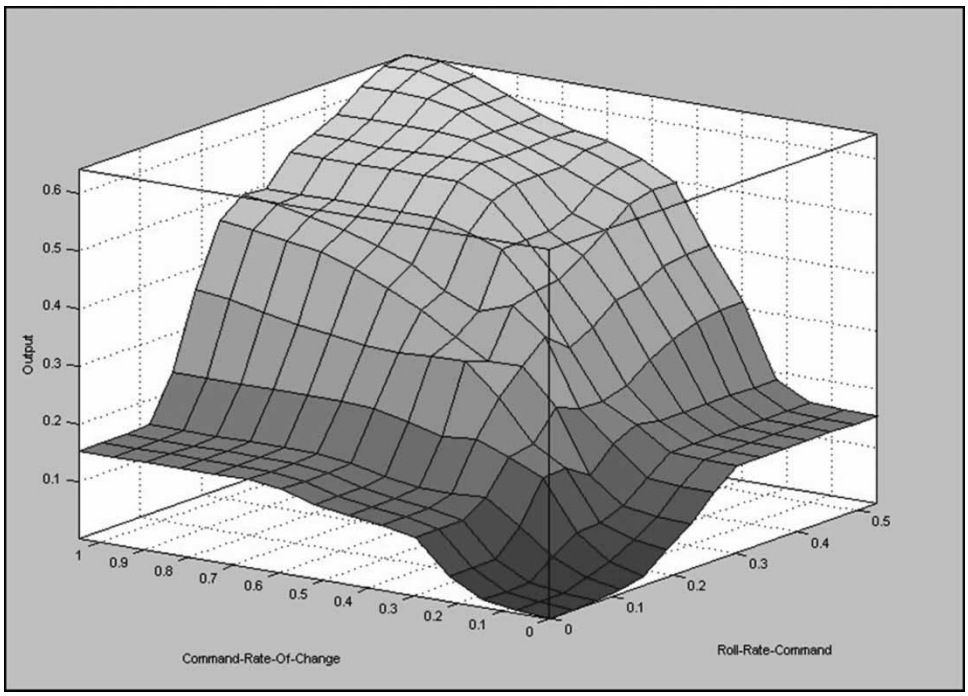

Fig. 11 Pitch controller optimal surface

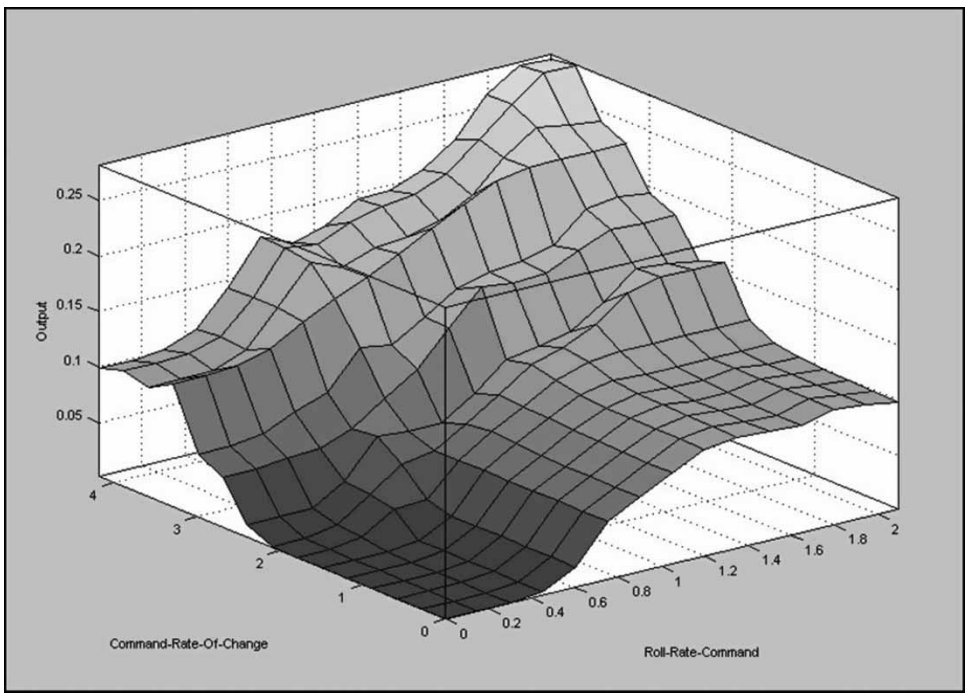

Fig. 12 Roll controller optimal surface

suggested that instability should be used to improve the system performance to small pilot demands. However, this was against the initial specification and it was ensured that the final surfaces did not include this attribute.

These developed surfaces were tested using step inputs with the results shown in Figs 14 to 16.

These results demonstrate that the new optimum fuzzy logic flight management controllers represent an improvement not only on their initial design but

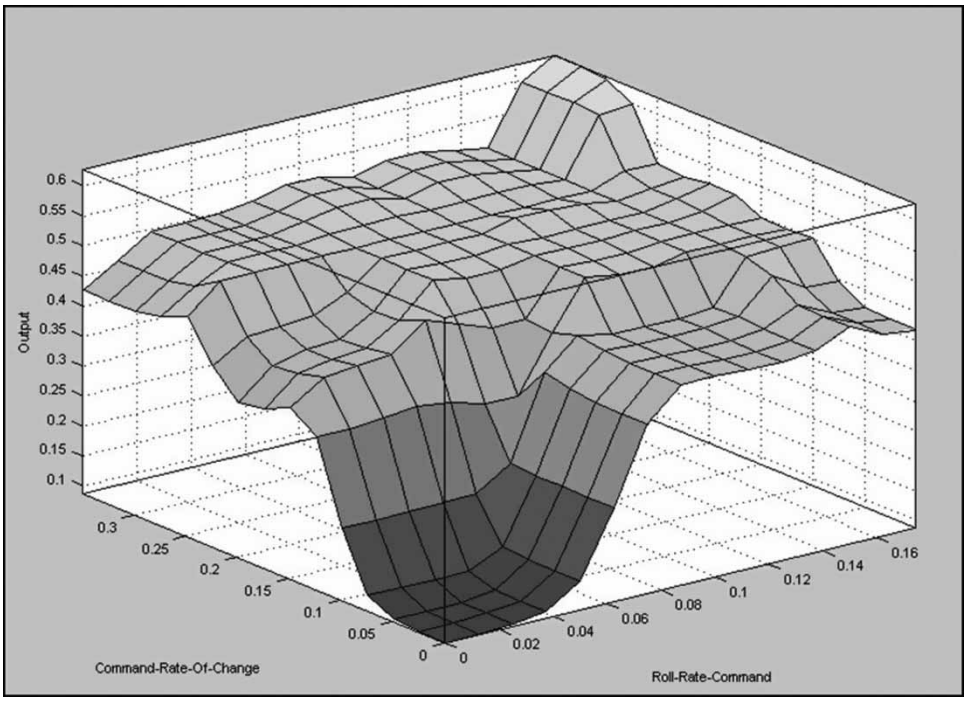

Fig. 13 Yaw controller optimal surface

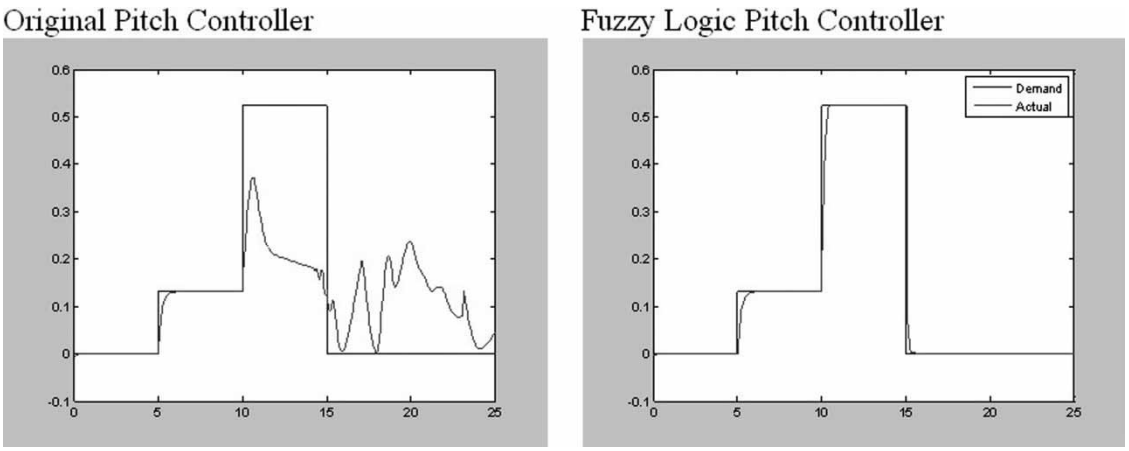

Fig. 14 Pitch controller step response

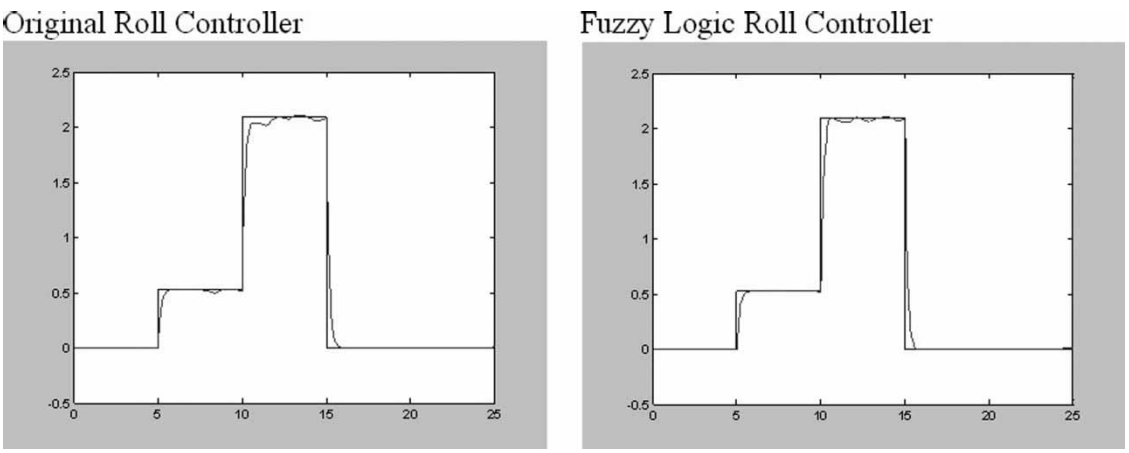

Fig. 15 Roll controller step response

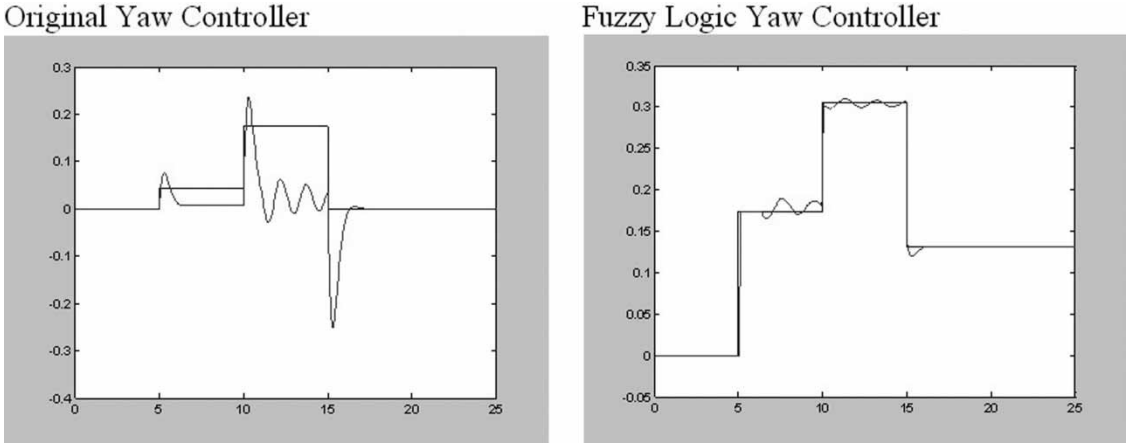

Fig. 16 Yaw controller step response

also on the initial system in terms of performance in response to pilot rate demands.

4 RULE BASE OPTIMIZATION

It was decided to investigate the optimization of the established rule base incorporating the memberships sets recently derived. This involved the creation of a new MOGA to utilize individuals that would construct new rules instead of membership classes. This and the development of a new routine to convert the rules into a suitable format for forming a fuzzy logic controller within Simulink were successfully achieved.

The final results of this analysis returned the original constructed rule-base even when starting from a set of random rules. Though this goes a small way to verifying that the original rule-base does represent an expert system, the results can also be explained by remembering that the membership functions were optimized to the original rule-base in the previously constructed MOGA. This would make it highly likely that when attempting to optimize a rule-base to these membership functions that the originally constructed rule-base would be returned.

5 TEST FLIGHT

In the designing of this improved flight controller, it has always been known that the aim was to improve aircraft manoeuvrability. This could be considered to be tested by assessing the following two points.

1. Tracking of pilot rate demands and actual rates achieved for all control surfaces while performing set manoeuvres.
2. Total pilot effort required to perform these set manoeuvres.

The designed test flight was constructed as follows.

1. Take-off and turn to a heading of due north while maintaining a 10° climb.
2. Increase climb to 45° until a height if 10 000 ft is achieved.
3. Level off aircraft and accelerate to Mach 1.5.
4. Perform two barrel rolls.
5. Reduce speed to Mach 0.75 and turn to a heading of due south.
6. Intercept automated drone for one rotation of the airfield.
7. Depart from interception on heading of 220° or 40° (NB. Runway in Cranfield Airfield Model runs from 40° to 220°).
8. Climb to 1000 ft.
9. Return to airfield and land safely.

This test flight procedure was performed for analysis only after suitable familiarization with the procedure had been achieved to prevent any influence from pilot skill affecting the results taken. The flight plan itself is constructed in a best effort to excite the important modes of operation of the system to aid analysis of performance. In particular, to analyse tracking of pilot rate demands, barrel roll, drone interception and interception departure are performed. The other trajectories which link the aerobatic manoeuvres are designed to assess pilot effort and tracking in a less dynamic environment. It was felt by the authors that this flight plan represents a good vehicle for assessment of the controller performance.

The pilot rate demands and aircraft responses were recorded through use of an inbuilt function of the real-time vehicle flight software. This allowed analysis of recorded real-time flight data with a pilot in the control loop; as apposed to the analysis of data produced through set inputs into a model.

6 FLIGHT CONTROLLER TRACKING

The relative tracking ability of each system was assessed through the calculation of the integral absolute

error between pilot rate demands and aircraft rate responses. The rate demand used for this analysis can be seen for the original controller in Fig. 17 and the augmented controller in Fig. 18.

Analysing by ITAE, the new controller has a ratio of 0.3842, a reduction of 61.58 per cent.

The mental effort required to fly an aeroplane, while meeting its mission objectives, are usually referred to as pilot compensation. If an aeroplane reacts either too slow or too fast to a pilot command, the pilot must compensate for the behaviour by 'adjusting his own gain' or 'leading' the airplane. Clearly a pilot should

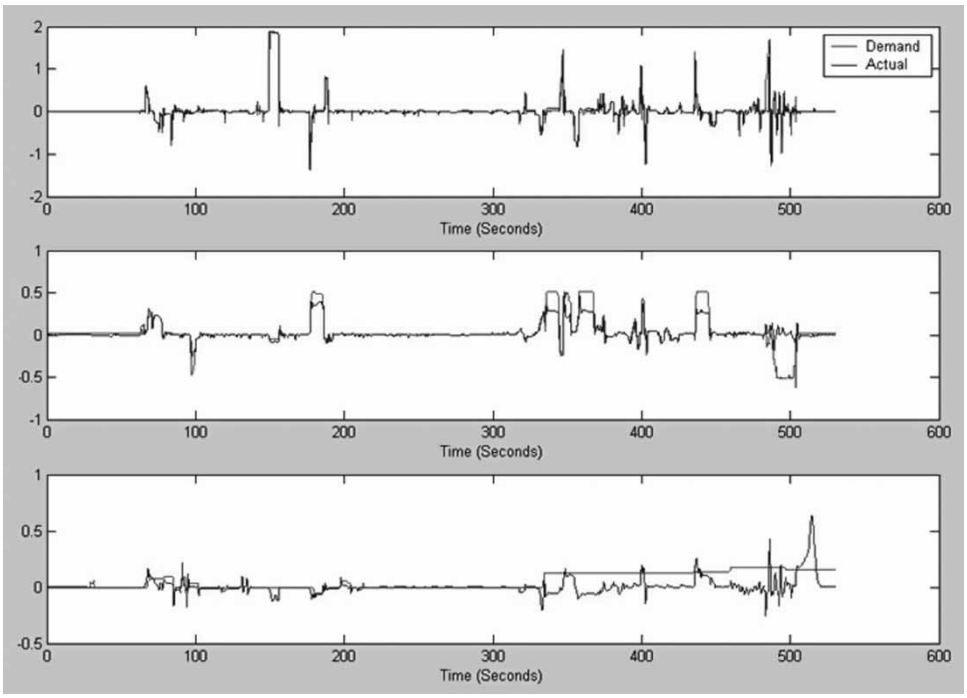

Fig. 17 Test flight performance of the original controller (roll, pitch, and yaw demands)

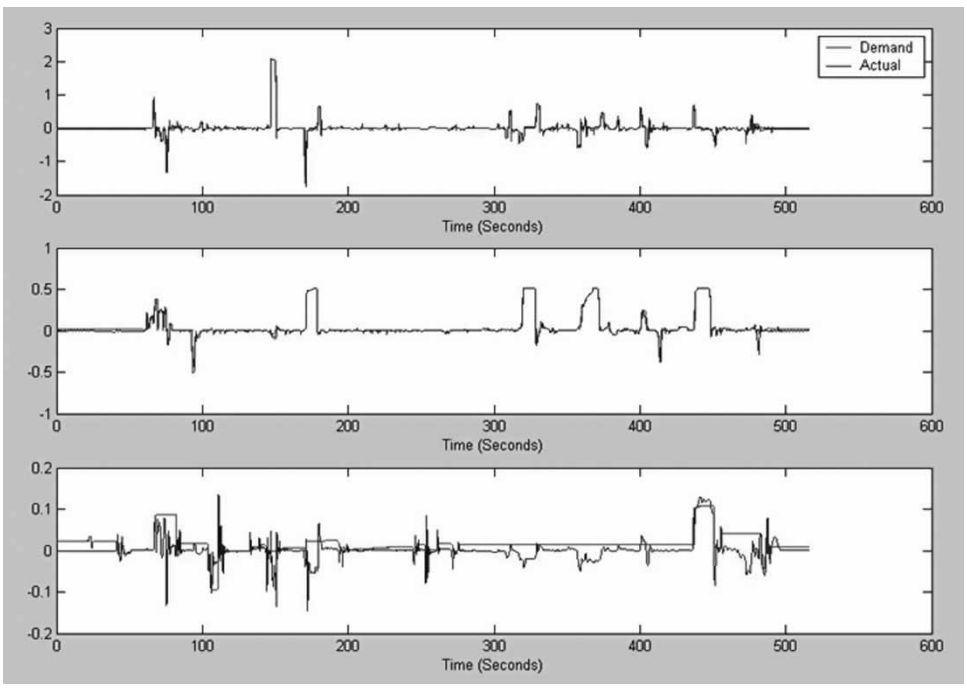

Fig. 18 Test flight performance of the fuzzy logic flight management controller (roll, pitch, and yaw demands)

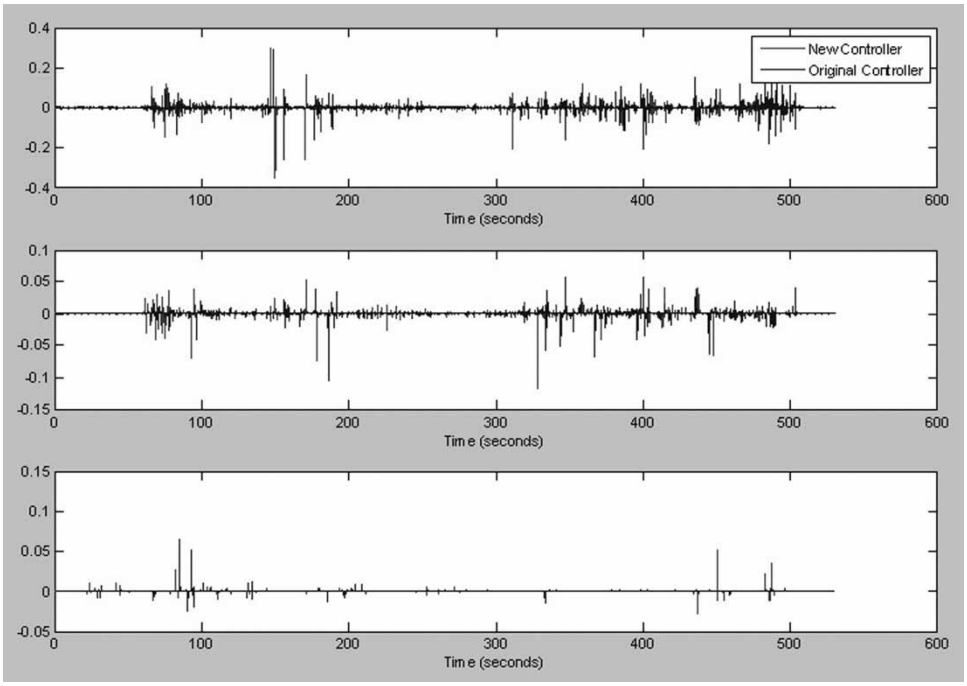

Fig. 19 Differential pilot inputs

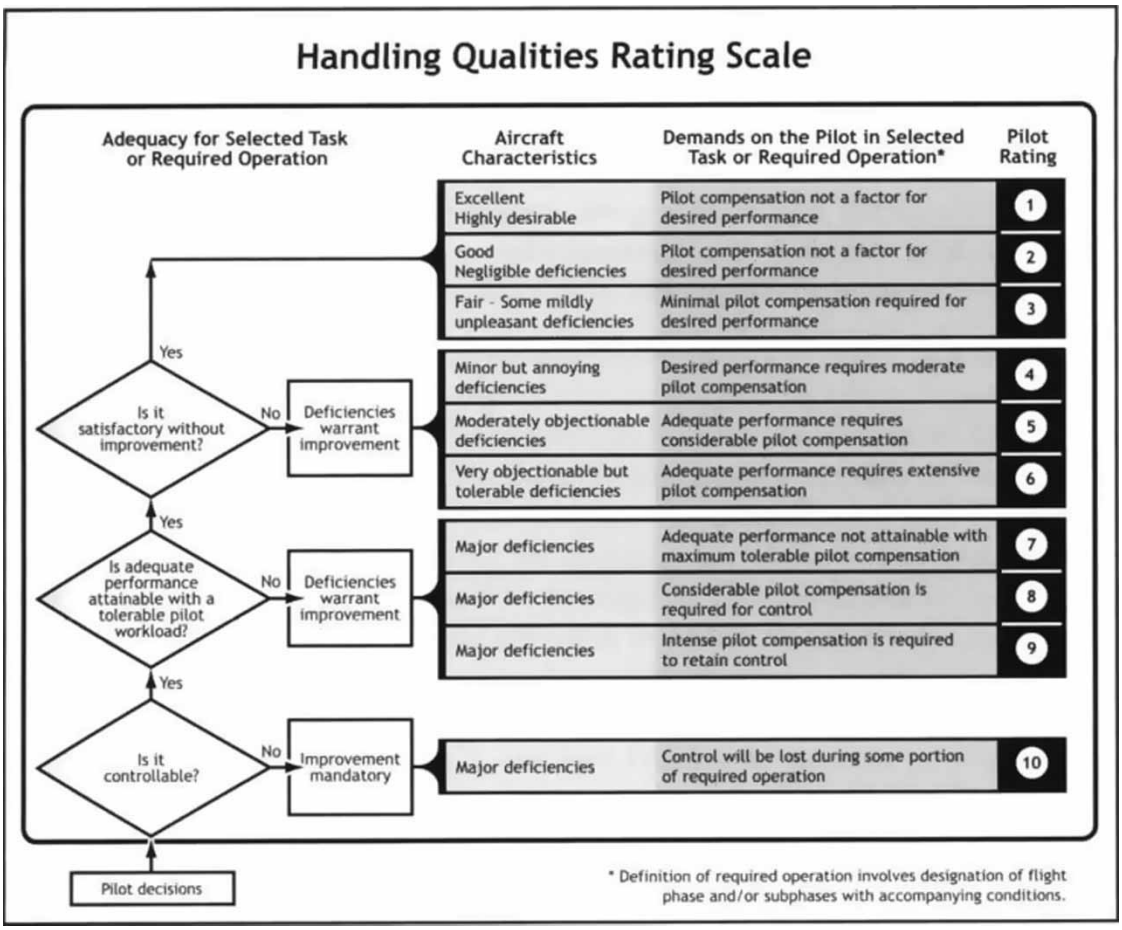

Fig. 20 Cooper–Harper scale

not have to excessively lead an aeroplane nor should they have too much or too little gain.

7 PILOT EFFORT

The pilot effort was analysed by both considering the absolute integral of pilot demands placed on the aircraft as well as the sum of the differential of these commands. The sum of the differential will provide an indication of the scale of commands made on the system; an indication of whether the pilot is forced to make large corrections due to over/under steer or many small corrections to keep the aircraft correctly trimmed. In which case the ratio of the new controller to old controller is 0.6215, an improvement of 37.85 per cent. An analysis of the differential of the pilot inputs to both systems to carry out the same test flight shows a 23.5 per cent reduction in the new controller when compared with the original F-16 flight controller. This would appear to suggest that the new controller required smaller corrections by the pilot to maintain the desired heading (Fig. 19).

7.1 Cooper–Harper rating scale

This is a set of criteria used by test pilots and flight test engineers to evaluate the handling qualities of aircraft during flight tests. The Cooper–Harper scale ranges from 1 to 10, with 1 indicating the best handling characteristics for an aircraft, and 10 the worst (Fig. 20). The criteria are evaluative and thus the scale is considered subjective. Therefore a rating is only valid when the aircraft is evaluated by an expert.

The F-16 flight controller was evaluated by a qualified pilot instructor. Although this individual was not familiar with the aircraft they were an expert in flying a number of aircraft types and their opinion was considered to be authoritative. The pilot rated the aircraft with the new fuzzy logic flight management controller as having the highest Cooper–Harper rating of 1.

8 CONCLUSION

A multi-objective design of a fuzzy logic augmented flight controller has been implemented and its performance analysed. In the initial sector of development, a standard fuzzy controller was designed, to augment the standard flight controller. The objective of this development was to increase the aircraft's aerobatic ability, reduce pilot fatigue, while still retaining the stability inherent in the core flight controller. The resulting augmented controller was found to be unsatisfactory in these aspects.

In order to improve the performance of the fuzzy logic controller, membership function tuning was carried out by MOGA. This methodology was selected due to MOGAs inherent capability to identify non-dominated solutions to highly complex objective functions, and also the capacity to run robustness measures as part of the optimization process.

The controller tuned by MOGA was designed to produce a MIMO roll, pitch and yaw controller with enhanced manoeuverability, while still retaining safety critical operation when combined with a standard inner-loop stabilizing controller.

The results demonstrated that the new fuzzy logic flight management controllers represent an improvement not only on the original uncompensated system but also on the original fuzzy logic design in response to pilot rate demands.

A separate study was performed with MOGA to investigate if it were possible to improve on the performance of the original fuzzy logic controller's rule-base. The final results of this analysis returned a rule-base identical to the original, confirming the validity of the original expert system.

Subsequent to the design process, the controller was incorporated into a 6DOF motion base real-time flight simulator and flight tested by a qualified pilot instructor. The controller was assessed in terms of pilot effort and reduction of pilot fatigue. The relative tracking ability of the controller was assessed and found to have an improvement of 61.58 per cent over the original flight controller without fuzzy augmentation. The pilot effort was found to be improved by 37.85 per cent due to smaller corrections necessary to maintain a desired heading or manoeuvre. The pilot instructor considered the handling characteristics of the aircraft to correspond to the highest level on the Cooper–Harper handling qualities scale.

In terms of the multi-objective design, the hard priority constraint of a maximum 9g loading on the airframe was not found possible to exceed, irrespective of aerobatic manoeuvres, and the effectiveness of the tracking and pilot effort ratios reflect the level of attainment of the objective function goals.

It has been shown that by the combination of a multi-objective optimization approach combined with fuzzy logic controller, the standard flight controller of an aircraft can be augmented to provide higher levels of manoeuverability, while retaining core stability, remaining within the airframe's safety limits, and reducing pilot error while improving tracking ability. This set of competing objectives present a highly appropriate problem for the application of multi-objective optimization, and have led to the development of an extremely useful novel design methodology.

© Authors 2010

REFERENCES

1 **Bryson Jr, A. E.** New concepts in control theory, 1959–1984. *J. Guid. Control Dyn.*, 1985, **8**(4), 417–425.
2 **Calise, A.** and **Rysdyk, R. T.** Nonlinear adaptive flight control using neural networks. *IEEE Control Syst. Mag.*, 1998, **18**(6), 14–25.
3 **Kim, B. S.** and **Calise, A. J.** Nonlinear flight control using neural networks. *J. Guid. Control Dyn.*, 1997, **20**(1), 26–33.
4 **Reigelsperger, W. C.** and **Banda, S. S.** Nonlinear simulation of a modified F-16 with full-envelope control laws. *Control Eng. Pract.*, 1998, **6**, 309–320.
5 **Ying, H.** A general technique for deriving analytical structure of fuzzy controllers using arbitrary trapezoidal input fuzzy sets. *Automatica*, 2004, **39**(7), 1171–1184.
6 **Li, Y., Sundararajan, N.,** and **Saratchandran, P.** Neuro-controlled design for nonlinear fighter aircraft maneuver using fully tuned RBF networks. *Automatica*, 2001, **37**, 1293–1301.
7 **Reiner, J., Balas, G. J.,** and **Garrard, W. L.** Flight control design using robust dynamic inversion and time scale separation. *Automatica*, 1996, **32**(11), 1493–1504.
8 **No, T. S., Min, B. M., Stone, R. H.,** and **Wong K. C.** Control and simulation of arbitrary flight trajectory-tracking. *Control Eng. Pract.*, 2005, **13**, 601–612.
9 **Keviczky, T.** and **Balas, G. J.** Receding horizon control of an F-16 aircraft: a comparative study. *Control Eng. Pract.*, 2000, **14**, 1023–1033.
10 **Kadmiry, B.** and **Driankov, D.** A fuzzy flight controller combining linguistic and model based fuzzy control. *Fuzzy Sets Syst.*, 2004, **146**, 313–347.
11 **Oosterom, M.** and **Babuska, R.** Design of a gain-scheduling mechanism for flight control laws by fuzzy clustering. *Control Eng. Pract.*, 2006, **14**, 769–781.
12 **Bates, D.** and **Hagstrom, M.** (Eds) Nonlinear analysis and synthesis techniques for aircraft control. In *Springer lecture notes in control and information sciences*, vol. 365, 2007 (Springer, London).
13 **Fung, E. H. K., Wong, Y. K., Liu, H. T.,** and **Li, Y. C.** Multiple specification controller design for F16 fighters. *J. Aircr. Eng. Aerosp. Technol.*, 2008, **80**(5), 510–515.
14 **Sonneveldt, L., Chu, Q. P.,** and **Mulder, J. A.** Nonlinear flight control using constrained adaptive backstepping. *J. Guid. Control Dyn.*, 2007, **30**(2), 322–336.
15 **Zhong, M. T.** and **An, J. W.** Application of intelligent computation to promote the automation of flight control design. In Proceedings of the IEEE International Conference on Intelligent Computation, Technology and Automation, 2008, vol. 1, pp. 991–994.
16 **Savran, A., Tasaltin, R.,** and **Becerikli, Y.** Intelligent adaptive nonlinear flight control for a high performance aircraft with neural networks. *ISA Trans.*, 2006, **45**(2), 225–247.
17 **Nguyen, L., Ogburn, M., Gilbert, W., Kibler, K., Brown, P.,** and **Deal, P.** Simulator study of stall/post stall characteristics of a fighter airplane with relaxed longitudinal static stability. NASA technical report 1538, 1979.
18 **Stevens, B.** and **Lewis, F.** *Aircraft control and simulation*, 2003 (Wiley, New York).
19 **Andry, A. N., Shapiro, E. Y.,** and **Chung, J. C.** Eigenstructure assignment for linear systems. *IEEE Trans. Aerosp. Electron. Syst.*, 1983, **19**(5), 711–724.
20 **Tischler, M. B.** (Ed.) *Advances in aircraft flight control*, 1996 (Taylor and Francis, London, UK).
21 **Gorzalczany, M. B.** *Computational intelligence systems and applications*, 2002, pp. 21–22 (Physica Verlag, Heidelberg, Germany).
22 **Baglio, S., Fortuna, L., Graziano, S.,** and **Muscato, G.** Membership function shape and the dynamic behaviour of fuzzy systems. *Int. J. Adapt. Control Signal Process.*, 2007, **8**(4), 369–377.
23 **Cooper, G. E.** and **Harper, R. P.** The use of pilot rating in the evaluation of aircraft handling qualities. NASA technical note TN-D5153, Washington, DC, USA, 1969.
24 **Cao, S. G., Reesand, N. W.,** and **Feng, G.** Fuzzy control of nonlinear continuous-time systems. In Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan, 1996, pp. 592–597.
25 **Castro, J. L.** Fuzzy logic controllers are universal approximators. *IEEE Trans. Syst. Man Cybern.*, 1998, **25**(4), 629–635.
26 **Fantuzzi, C.** and **Rovatti, R.** On the approximation capabilities of the homogeneous Takagi-Sugeno model. In Proceedings of the 5th IEEE International Conference on Fuzzy Systems, 1996, pp. 1067–1072.
27 **Belarbi, K., Titel, F., Bourebia, W.,** and **Benmahammed, K.** Design of Mamdani fuzzy logic controllers with rule base minimisation using genetic algorithms. *Eng. Appl. Artif. Intell.*, 2005, **18**(7), 875–880.
28 **Stewart, P., Stone, D. A.,** and **Fleming, P. J.** Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *IFAC J. Eng. Appl. Artif. Intell.*, 2004, **70**(3), 275–284.
29 **Fonseca, C. M.** and **Fleming, P. J.** Genetic algorithms for multiobjective optimisation: formulation, discussion and generalisation. In *Genetic algorithms*: Proceedings of the Fifth International Conference, San Mateo, California, 1993, pp. 416–423 (Morgan Kaufmann, San Fransisco).
30 **Fonseca, C. M.** and **Fleming, P. J.** Multiobjective optimisation and multiple constraint handling with evolutionary algorithms – part 1: a unified formulation and part 2: application example. *IEEE Trans. Syst. Man Cybern. A, Syst. Humans*, 1998, **28**(1), 26–37, 38–47.
31 **Purshouse, R. C.** and **Fleming, P. J.** The multiobjective genetic algorithm applied to benchmark problems – an analysis. Research report no. 796, Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 2001.
32 **Goldberg, D. E.** *Genetic algorithms in search, optimisation and machine learning*, 1989 (Addison-Wesley Professional, Indianapolis, USA).
33 **Wu, D.** and **Tan, W. W.** Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers. *Eng. Appl. Artif. Intell.*, 2006, **19**, 829–841.

APPENDIX

MOGA parameters

Population size: 40
Number of decision variables: 34

Number of objectives: 4 per axis
Number of immigrants per generation: 6
Coding: grey, 20 bits per decision variable, except where varied
Selection: stochastic universal sampling
Recombination: single-point binary crossover, probability = 0.7
Mutation: element-wise bit-flipping, expectation of 1 bit per chromosome
Generational gap: zero
Random injection: 2 random chromosomes per generation
Elitism: none
Fitness assignment: [**29**] multi-objective ranking. Transformation from rank to fitness using linear fitness assignment with rank-wise averaging.
External population: off-line storage of non-dominated solutions.
Fitness sharing: (parameter-less) Epanechnikov fitness sharing [**29**] implemented in criterion space.
Mating restriction implemented: distance set to the niche size parameter found by the Epanechnikov sharing algorithm.